

| | |
|--------------------------|--|
| Assignment Title: | Write a Python program to store marks scored in subject “Fundamental of Data Structure” by N students in the class. Write functions to compute following: a) The average score of class b) Highest score and lowest score of class c) Count of students who were absent for the test d) Display mark with highest frequency |
| Assignment No.: | A-01 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
def average(lst):  
    sum=0  
    count=0  
    for i in range(len(lst)):  
        if lst[i]!=-1:  
            sum+=lst[i]  
            count+=1  
        avg=sum/count  
    print("\nAverage marks are : ",avg)
```

```
def maxmarks(lst):  
    maxx=lst[0]  
    for i in range(len(lst)):  
        if(lst[i]>maxx):  
            maxx=lst[i]  
    print("\nHighest marks are : ",maxx)
```

```
def minmarks(lst):  
    minn=lst[0]
```

```
for i in range(len(lst)):
    if((lst[i]<minn) and (lst[i]!=-1)):
        minn=lst[i]
print("\nLowest marks are : ",minn)
```

```
def absent(lst):
    count=0
    for i in range(len(lst)):
        if(lst[i]==-1):
            count+=1
    print("\nNo. of absent students is : ",count)
```

```
def maxfreq(lst):
    max1=0
    for i in lst:
        count=0
        for j in lst:
            if i==j and j!=-1:
                count+=1
        if count>=max1:
            max1=count
            a=i
    print("\nMaximum frequency is",max1,"of",a)
```

```
def main():
    mark=[]
    n=int(input("\nEnter total no. of students : "))
    for i in range(n):
        a=int(input("\nEnter marks of student %2d :"%(i+1)))
        mark.append(a)
    while True:
        print("\n*****Menu*****")
        print("\n1. The average score of the class")
        print("\n2. Highest and lowest score of the class")
        print("\n3. Count of students who were absent for the test")
        print("\n4. Display marks with highest frequency")
        print("\n5. Exit")
        ch=int(input("\nEnter choice : "))
        if ch==1:
            average(mark)
        elif ch==2:
```

```
    maxmarks(mark)
    minmarks(mark)
elif ch==3:
    absent(mark)
elif ch==4:
    maxfreq(mark)
elif ch==5:
    print("\nExiting program....")
    break
else:
    print("\nIncorrect choice")
main()
```

| |
|------------------------|
| Program Output: |
|------------------------|

Enter total no. of students : 10

Enter marks of student 1 : 10

Enter marks of student 2 : 20

Enter marks of student 3 : -1

Enter marks of student 4 : 25

Enter marks of student 5 : 29

Enter marks of student 6 : -1

Enter marks of student 7 : 20

Enter marks of student 8 : 30

Enter marks of student 9 : 10

Enter marks of student 10 : 20

*****Menu*****

1. The average score of the class
2. Highest and lowest score of the class
3. Count of students who were absent for the test
4. Display marks with highest frequency
5. Exit

Enter choice : 1

Average marks are : 20.5

*****Menu*****

1. The average score of the class
2. Highest and lowest score of the class
3. Count of students who were absent for the test
4. Display marks with highest frequency
5. Exit

Enter choice : 2

Highest marks are : 30

Lowest marks are : 10

*****Menu*****

1. The average score of the class
2. Highest and lowest score of the class
3. Count of students who were absent for the test

4. Display marks with highest frequency

5. Exit

Enter choice : 3

No. of absent students is : 2

*****Menu*****

1. The average score of the class

2. Highest and lowest score of the class

3. Count of students who were absent for the test

4. Display marks with highest frequency

5. Exit

Enter choice : 4

Maximum frequency is 3 of 20

*****Menu*****

1. The average score of the class

2. Highest and lowest score of the class

3. Count of students who were absent for the test

4. Display marks with highest frequency

5. Exit

Enter choice : 5

Exiting program....

| | |
|--------------------------|---|
| Assignment Title: | Write a Python program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following: D 100 W 200 (Withdrawal is not allowed if balance is going negative. Write functions for withdraw and deposit) D means deposit while W means withdrawal. Suppose the following input is supplied to the program: D 300 D 300 W 200 D 100 Then, the output should be: 500 |
| Assignment No.: | A-02 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

| |
|----------------------|
| Program Code: |
|----------------------|

```
def deposit():
    log=input("\nEnter transaction log : ")
    account=(log.split())
    if(account[0]=='D'):
        return account[1]
    else:
        print("\nEnter correct transaction log")

def withdraw():
    log=input("\nEnter transaction log : ")
    account=(log.split())
    if(account[0]=='W'):
        return account[1]
    else:
        print("\nEnter correct transaction log")
```

```
def main():
    balance=0
    while True:
        print("\n*****Menu*****")
        print("\n1. Deposit money")
        print("\n2. Withdraw money")
        print("\n3. Exit")
        ch=int(input("\nEnter choice : "))
        if ch==1:
            a=deposit()
            balance+=int(a)
            print("\nCurrent balance : ",balance)
        elif ch==2:
            b=withdraw()
            if(balance>=int(b)):
                balance=balance-int(b)
                print("\nCurrent balance : ",balance)
            else:
                print("\nInsufficient balance")
        elif ch==3:
            print("Exiting program.....")
            break
        else:
            print("\nIncorrect choice")

main()
```

Program Output:

```
*****Menu*****
```

```
1. Deposit money
```

```
2. Withdraw money
```

```
3. Exit
```

```
Enter choice : 1
```

Enter transaction log : D 300

Current balance : 300

*****Menu*****

1. Deposit money
2. Withdraw money
3. Exit

Enter choice : 1

Enter transaction log : D 300

Current balance : 600

*****Menu*****

1. Deposit money
2. Withdraw money
3. Exit

Enter choice : 2

Enter transaction log : W 200

Current balance : 400

*****Menu*****

1. Deposit money
2. Withdraw money
3. Exit

Enter choice : 1

Enter transaction log : D 100

Current balance : 500

*****Menu*****

1. Deposit money

2. Withdraw money

3. Exit

Enter choice : 3

Exiting program.....

| | |
|--------------------------|---|
| Assignment Title: | Write a Python program to compute following operations on String: a) To display word with the longest length b) To determines the frequency of occurrence of particular character in the string c) To check whether given string is palindrome or not d) To display index of first appearance of the substring e) To count the occurrences of each word in a given string |
| Assignment No.: | A-03 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
def longestword():
    str=input("\nEnter string : ")
    long = ""
    i = 0
    while(i < len(str)):
        str1 = ""
        if(i != len(str)):
            while(str[i] == " "):
                i=i+1
            while(str[i] != " "):
                str1 = str1 + str[i]
                i=i+1
            if(i == len(str)):
                break
        if(len(str1)>len(long)):
            long = str1
    print("\nLongest word is : ",long)
```

```
def character():
```

```
str1 = input("\nEnter string : ")
char = input("\nEnter character : ")
counter = 0
for i in range(len(str1)):
    if char == str1[i]:
        counter = counter + 1
print("\nThe occurrence of ",char," is : ",counter)
```

```
def palindrome():
    str2 = input("\nEnter string : ")
    lenstr2 = len(str2)
    j = lenstr2 - 1
    flag = 0
    for i in range(int(lenstr2 / 2 + 1)):
        if (str2[i] == str2[j]):
            j -= 1
            flag = 1
        else:
            flag = 0
            break
    if (flag == 1):
        print("\nEnter string is Palindrome")
    else:
        print("\nEnter string is not Palindrome")
```

```
def substring():
    str1 = input("\nEnter string : ")
    sub1 = input("\nEnter substring : ")
    m = len(sub1)
    n = len(str1)
    if(n>=m):
        for i in range((n-m+1)):
            I=1
            for j in range(m):
                if(str1[i+j]!=sub1[j]):
                    I=0
                    break
            if(I==1):
                print("\nIndex of the substring is : ",i)
                break
```

```
    if(I==0):
        print("\nNot found")

def occword():
    str = input("\nEnter the string : ")
    i=0
    word_array=[ ]
    count=[]
    while(i<len(str)):
        word=""
        while(str[i]!=" "):
            word+=str[i]
            i=i+1
        if(i==len(str)):
            break
        if(i!=len(str)):
            while(str[i]==" "):
                i=i+1
        if(len(word_array)==0):
            word_array.append(word)
            count.append(1)
        else:
            flag=1
            for j in range(len(word_array)):
                if(word_array[j]==word):
                    count[j]+=1
                    flag=0
                    break
            if(flag==1):
                word_array.append(word)
                count.append(1)
        for i in range(len(word_array)):
            print("%10s : %d"%(word_array[i],count[i]))

while True:
    print("\n*****MENU*****")
    print("\n1. To display word with the longest length")
    print("\n2. To determine the frequency of occurrence of particular character in the string")
    print("\n3. To check whether the given string is palindrome or not")
    print("\n4. To display index of first appearance of the substring")
```

```
print("\n5. To count the occurrence of each word in a given string")
print("\n6. Exit")
ch = int(input("\nEnter your choice : "))
if (ch == 1):
    longestword()
elif (ch == 2):
    character()
elif (ch == 3):
    palindrome()
elif (ch == 4):
    substring()
elif (ch == 5):
    occword()
elif (ch == 6):
    print("\nExiting program....")
    break
```

Program Output:

*****MENU*****

1. To display word with the longest length
2. To determine the frequency of occurrence of particular character in the string
3. To check whether the given string is palindrome or not
4. To display index of first appearance of the substring
5. To count the occurrence of each word in a given string
6. Exit

Enter your choice : 1
Enter string : Data Structures Lab
Longest word is : Structures

*****MENU*****

1. To display word with the longest length
2. To determine the frequency of occurrence of particular character in the string
3. To check whether the given string is palindrome or not
4. To display index of first appearance of the substring
5. To count the occurrence of each word in a given string
6. Exit

Enter your choice : 2

Enter string : Structures

Enter character : r

The occurrence of r is : 2

*****MENU*****

1. To display word with the longest length
2. To determine the frequency of occurrence of particular character in the string
3. To check whether the given string is palindrome or not
4. To display index of first appearance of the substring
5. To count the occurrence of each word in a given string
6. Exit

Enter your choice : 3

Enter string : madam

Entered string is Palindrome

*****MENU*****

1. To display word with the longest length
2. To determine the frequency of occurrence of particular character in the string

3. To check whether the given string is palindrome or not
4. To display index of first appearance of the substring
5. To count the occurrence of each word in a given string
6. Exit

Enter your choice : 4

Enter string : Data Structures Lab

Enter substring : ruct

Index of the substring is : 7

*****MENU*****

1. To display word with the longest length
2. To determine the frequency of occurrence of particular character in the string
3. To check whether the given string is palindrome or not
4. To display index of first appearance of the substring
5. To count the occurrence of each word in a given string
6. Exit

Enter your choice : 5

Enter the string : Data Structures Lab comes under Fundamentals of Data Structures

Data : 2

Structures : 2

Lab : 1

comes : 1

under : 1

Fundamentals : 1

of : 1

*****MENU*****

1. To display word with the longest length
2. To determine the frequency of occurrence of particular character in the string
3. To check whether the given string is palindrome or not
4. To display index of first appearance of the substring
5. To count the occurrence of each word in a given string
6. Exit

Enter your choice : 6

Exiting program....

| | |
|--------------------------|---|
| Assignment Title: | Write a Python program to maintain club members, sort on roll numbers in ascending order. Write function "Ternary_Search" to search whether particular student is member of club or not. Ternary search is modified binary search that divides array into 3 halves instead of two. |
| Assignment No.: | A-04 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
def accept_array(A):
    n = int(input("\nEnter the total number of the students : "))
    for i in range(n):
        a=(int(input("\nEnter roll number of student %d: "%(i+1))))
        A.append(a)
    print("\nStudent information collected successfully")
    return n
def display(A,n):
    if (n == 0):
        print("\nData not found")
    else:
        print("\nStudent array : ", end=" ")
        for i in range(n):
            print("%d"%A[i],end=' ')
        print("\n")

def ternary_search(A, l, r, key):
    if (r >= 1):
        mid1 = l + int((r - l) / 3)
        mid2 = r - int((r - l) / 3)
```

```
    if (A[mid1] == key):
        return mid1
    if (A[mid2] == key):
        return mid2
    if (key < A[mid1]):
        return ternary_search(A, l, mid1 - 1, key)
    elif (key > A[mid2]):
        return ternary_search(A, mid2 + 1, r, key)
    else:
        return ternary_search(A, mid1 + 1, mid2 - 1, key)
return -1
```

```
def main():
    A = []
    while True:
        print("\n*****MENU*****")
        print("\n1 : Accept student information ")
        print("\n2 : Execute Ternary search ")
        print("\n3 : Exit ")
        ch = int(input("\nEnter choice : "))
        if (ch == 1):
            A = []
            n = accept_array(A)
            display(A, n)
        elif (ch == 2):
            X = int(input("\nEnter the roll number to be searched : "))
            flag = ternary_search(A, 0, n-1, X)
            if (flag == -1):
                print("\nRoll no.",X,"is not a member of the club.")
            else:
                print("\nRoll no.",X,"is a member of the club stored at the location ", flag + 1)
        elif (ch == 3):
            print("\nExiting program....")
            break
        else:
            print("\nPlease enter the correct choice")

main()
```

| |
|------------------------|
| Program Output: |
|------------------------|

*****MENU*****

1. Accept student information

2. Execute ternary search

3. Exit

Enter your choice : 1

Enter the total number of the students : 10

Enter roll number of student 1 : 20

Enter roll number of student 2 : 25

Enter roll number of student 3 : 31

Enter roll number of student 4 : 38

Enter roll number of student 5 : 43

Enter roll number of student 6 : 47

Enter roll number of student 7 : 55

Enter roll number of student 8 : 64

Enter roll number of student 9 : 68

Enter roll number of student 10 : 72

Student information collected successfully

Student array : 20 25 31 38 43 47 55 64 68 72

*****MENU*****

1. Accept student information
2. Execute ternary search
3. Exit

Enter your choice : 2

Enter the roll number to be searched : 64

Roll no. 64 is a member of the club stored at the location 8

*****MENU*****

1. Accept student information
2. Execute ternary search
3. Exit

Enter your choice : 3

Exiting program....

| | |
|--------------------------|--|
| Assignment Title: | Write a Python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using a) Selection Sort b) Bubble sort and display top five scores. |
| Assignment No.: | A-05 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
def array(A):
    n=int(input("\nEnter the total no. of students : "))

    for i in range(n):
        x=int(input("\nEnter first year percentage of student %d : " %(i+1)))
        A.append(x)
    print("\nStudent information accepted successfully\n")
    return n

def display(A):
    n=len(A)
    if(n==0):
        print("\nNo records in the database")
    else:
        print("\nStudents array : ",end=' ')
        for i in range(n):
            print("%d" %A[i], end=' ')
        print(" ")

def Selection_sort(A):
    n=len(A)
```

```
for i in range(n-1):
    min=i
    for j in range(i+1,n):
        if(A[j]<A[min]):
            min=j
    temp=A[i]
    A[i]=A[min]
    A[min]=temp
```

```
def bubble_sort(A):
    n=len(A)
    for i in range(1,n):
        for j in range(n-i):
            if(A[j]>A[j+1]):
                temp=A[j]
                A[j]=A[j+1]
                A[j+1]=temp
```

```
def main():
    A=[]
    while True:
        print("*****MENU*****")
        print("\n1. Accept FE marks of students")
        print("\n2. Execute selection sort in ascending order")
        print("\n3. Execute bubble sort in ascending order and Display top five scores")
        print("\n4. Exit")
        choice=int(input("\nEnter your choice : "))
        if(choice==1):
            A=[]
            array(A)
            display(A)
        elif(choice==2):
            print("\nMarks before sorting")
            display(A)
            Selection_sort(A)
            print("\nMarks after sorting")
            display(A)
        elif(choice==3):
            n=len(A)
            print("\nMarks before sorting")
```

```
display(A)
bubble_sort(A)
print("\nMarks after sorting")
display(A)
if(len(A)>=5):
    print("\nTop Five Scores :")
    n=len(A)
    if(len(A)>=5):
        B=[]
        for i in range(n-5,n):
            x=A[i]
            B.append(x)
        s=len(B)
        m=int(s/2)
        for j in range(m):
            temp=B[j]
            B[j]=B[s-j-1]
            B[s-j-1]=temp
        for j in range(s):
            print("\t %.2f"%B[j])
    else:
        print("\nTop Five scores:")
        s=len(A)
        m=int(s/2)
        for j in range(m):
            temp=A[j]
            A[j]=A[s-j-1]
            A[s-j-1]=temp
        for j in range(s):
            print("\t %.2f"%A[j])
elif(choice==4):
    print("\nExiting program.....")
    break
else:
    print("\nIncorrect choice")

main()
```

| |
|------------------------|
| Program Output: |
|------------------------|

*****MENU*****

1. Accept FE marks of students
2. Execute selection sort in ascending order
3. Execute bubble sort in ascending order and Display top five scores
4. Exit

Enter your choice : 1

Enter the total no. of students : 10

Enter first year percentage of student 1 : 65

Enter first year percentage of student 2 : 72

Enter first year percentage of student 3 : 78

Enter first year percentage of student 4 : 83

Enter first year percentage of student 5 : 89

Enter first year percentage of student 6 : 92

Enter first year percentage of student 7 : 56

Enter first year percentage of student 8 : 75

Enter first year percentage of student 9 : 86

Enter first year percentage of student 10 : 91

Student information accepted successfully

Students array : 65 72 78 83 89 92 56 75 86 91

*****MENU*****

1. Accept FE marks of students
2. Execute selection sort in ascending order
3. Execute bubble sort in ascending order and Display top five scores
4. Exit

Enter your choice : 2

Marks before sorting

Students array : 65 72 78 83 89 92 56 75 86 91

Marks after sorting

Students array : 56 65 72 75 78 83 86 89 91 92

*****MENU*****

1. Accept FE marks of students
2. Execute selection sort in ascending order
3. Execute bubble sort in ascending order and Display top five scores
4. Exit

Enter your choice : 1

Enter the total no. of students : 10

Enter first year percentage of student 1 : 65

Enter first year percentage of student 2 : 72

Enter first year percentage of student 3 : 78

Enter first year percentage of student 4 : 83

Enter first year percentage of student 5 : 89

Enter first year percentage of student 6 : 92

Enter first year percentage of student 7 : 56

Enter first year percentage of student 8 : 75

Enter first year percentage of student 9 : 86

Enter first year percentage of student 10 : 91

Student information accepted successfully

Students array : 65 72 78 83 89 92 56 75 86 91

*****MENU*****

1. Accept FE marks of students
2. Execute selection sort in ascending order
3. Execute bubble sort in ascending order and Display top five scores
4. Exit

Enter your choice : 3

Marks before sorting

Students array : 65 72 78 83 89 92 56 75 86 91

Marks after sorting

Students array : 56 65 72 75 78 83 86 89 91 92

Top Five Scores :

92.00

91.00

89.00

86.00

83.00

*****MENU*****

1. Accept FE marks of students

2. Execute selection sort in ascending order

3. Execute bubble sort in ascending order and Display top five scores

4. Exit

Enter your choice : 4

Exiting program.....

| | |
|--------------------------|---|
| Assignment Title: | Write a Python program to store 12th class percentage of students in array. Write function for sorting array of floating point numbers in ascending order using bucket sort and display top five scores. |
| Assignment No.: | A-06 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
def accept_array(A):
    n=int(input("\nEnter the total no. of students : "))
    for i in range(n):
        x=int(input("\nEnter percentage of class 12th student %d :"%(i+1)))
        A.append(x)
    print("\nStudent information accepted successfully");

def display_array(A):
    n=len(A)
    if(n==0):
        print("\nNo records in the database")
    else:
        print("\nStudent array : ",end=' ')
        for i in range(n):
            print("%d "%A[i],end=' ')
        print("\n");

def find_max_digit(A,n):
    max_element=max(A)
    count=0
    while(max_element>0):
        count=count+1
        max_element=max_element//10
```

```
    return count
```

```
def combinebucket(A,B):  
    c=0  
    for i in range(10):  
        for j in range(len(B[i])):  
            A[c]=B[i][j]  
            c=c+1
```

```
def initialize_bucket(B):  
    for i in range(10):  
        T=[]  
        B.append(T)
```

```
def Bucket_sort(A,n):  
    shift=1  
    keysize=find_max_digit(A,n)  
    for loop in range(keysize):  
        B=[]  
        initialize_bucket(B)  
        for i in range(n):  
            b_no=int((A[i]/shift)) % 10  
            B[b_no].append(A[i])  
        combinebucket(A,B)  
        shift=shift*10
```

```
def main():  
    A=[]  
    while True:  
        print("\n*****MENU*****")  
        print("\n1. Accept students FE marks")  
        print("\n2. Bucket sort and Display top five scores")  
        print("\n3. Exit")  
        choice=int(input("\nEnter your choice : "))  
        if(choice==3):  
            print("\nExiting program....")  
            break  
        elif(choice==1):  
            A=[]  
            accept_array(A)
```

```
        display_array(A)
    elif(choice==2):
        print("\nMarks before sorting")
        display_array(A)
        n=len(A)
        Bucket_sort(A,n)
        print("\nMarks after sorting")
        display_array(A)
        if(n>=5):
            print("\nTop five scores : ")
            for i in range(n-1,n-6,-1):
                print("\t%d"%A[i])
        else:
            print("\nTop five scores : ")
            for i in range(n-1,-1,-1):
                print("\t%d"%A[i])
    else:
        print("\nIncorrect choice")

main()
```

Program Output:

*****MENU*****

1. Accept students FE marks
2. Bucket sort and Display top five scores
3. Exit

Enter your choice : 1

Enter the total no. of students : 10

Enter percentage of class 12th student 1 : 98

Enter percentage of class 12th student 2 : 82

Enter percentage of class 12th student 3 : 75

Enter percentage of class 12th student 4 : 68

Enter percentage of class 12th student 5 : 72

Enter percentage of class 12th student 6 : 88

Enter percentage of class 12th student 7 : 85

Enter percentage of class 12th student 8 : 91

Enter percentage of class 12th student 9 : 79

Enter percentage of class 12th student 10 : 90

Student information accepted successfully

Student array : 98 82 75 68 72 88 85 91 79 90

*****MENU*****

1. Accept students FE marks

2. Bucket sort and Display top five scores

3. Exit

Enter your choice : 2

Marks before sorting

Student array : 98 82 75 68 72 88 85 91 79 90

Marks after sorting

Student array : 68 72 75 79 82 85 88 90 91 98

Top five scores :

98

91

90

88

85

*****MENU*****

1. Accept students FE marks

2. Bucket sort and Display top five scores

3. Exit

Enter your choice : 3

Exiting program.....

| | |
|--------------------------|--|
| Assignment Title: | Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. First node is reserved for president of club and last node is reserved for secretary of club. Write C++ program to maintain club member's information using singly linked list. Store student PRN and Name. Write functions to: a) Add and delete the members as well as president or even secretary. b) Compute total number of members of club c) Display members d) Two linked lists exist for two divisions. Concatenate two lists. |
| Assignment No.: | A-07 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
#include <iostream>
#include <string.h>
using namespace std;
```

```
class SLL;
class dnode
{
    int div;
    int prn;
    char name[20];
    dnode *next;
```

```
    friend SLL;
```

```
public:
```

```

dnode()
{
    next = NULL;
    div = prn = 0;
}
dnode(int d, int p, char Name[])
{
    div = d;
    prn = p;
    next = NULL;
    strcpy(name, Name);
}
};

class SLL
{
public:
    dnode *head;
    dnode *insert;
    dnode *end;

    void create();
    void print();
    void insertMember();
    void deleteMember();
    void count();
    void mergeList();
    void recursionPrint(dnode *temp);

    SLL()
    {
        head = NULL;
        insert = NULL;
        end = NULL;
    }
} list1;

void SLL::create()
{
    int n, d, p;
```

```
char Name[20];

cout << "Enter the number of students: ";
cin >> n;

cout << "\nEnter the name, division and PRN of President: \n";
cin >> Name >> d >> p;

head = new dnode(d, p, Name);
end = head;

cout << "\nEnter the name, division and PRN of Member : \n";
for (int i = 1; i < n - 1; i++)
{
    cin >> Name >> d >> p;
    end->next = new dnode(d, p, Name);
    end = end->next;
}
insert = end;

cout << "\nEnter the name, division and PRN of Secretary : \n";
cin >> Name >> d >> p;
end->next = new dnode(d, p, Name);
end = end->next;
end->next = NULL;
}

void SLL::insertMember()
{
    int d, p;
    char Name[20];
    cout << "\nEnter the name, division and PRN of member to insert: \n";
    cin >> Name >> d >> p;
    insert->next = new dnode(d, p, Name);
    insert = insert->next;
    insert->next = end;
}

void SLL::deleteMember()
{
```

```
int searchPRN;
dnode *remove;
cout << "Enter the PRN of member to delete: ";
cin >> searchPRN;
if (searchPRN == head->prn)
{
    remove = head;
    head = head->next;
    delete remove;
}
else
{
    for (dnode *temp = head; temp != insert; temp = temp->next)
    {
        if (temp->next->prn == searchPRN)
        {
            if (temp->next == insert)
                insert = temp;
            remove = temp->next;
            temp->next = temp->next->next;
            delete remove;

            if (temp == insert)
                break;
        }
    }
}
if (searchPRN == end->prn)
{
    remove = end;
    insert->next = NULL;
    end = insert;
    delete remove;
}
}

void SLL::count()
{
    int count = 0;
    for (dnode *temp = head; temp != NULL; temp = temp->next)
```

```
        count++;
    cout << "\nTotal no. of students : " << count;
}

void SLL::print()
{
    cout << "\n\nNAME\tDIV\tPRN\n";
    for (dnode *temp = head; temp != NULL; temp = temp->next)
        cout << temp->name << "\t" << temp->div << "\t" << temp->prn << "\n";
    cout << "\nPresident is: " << head->name;
    cout << "\nSecretary is: " << end->name << "\n";
}

void SLL::mergeList()
{
    SLL list2;
    cout << "\nEnter the contents for second list: \n";
    list2.create();
    list1.end->next = list2.head;
    list1.end = list2.end;
}

void SLL::recursionPrint(dnode *temp)
{
    if (temp == NULL)
        return;
    else
        recursionPrint(temp->next);
    cout << temp->name << "\t" << temp->div << "\t" << temp->prn << "\n";
}

int main()
{
    int choice;
    do
    {
        cout << "\n*****MENU*****";
        cout << "\n1. Create List. \n2. Insert Member \n3. Delete Member \n4. Print List \n5. Merge
List \n6. Reverse List \n7. Count \n8. Exit";
        cout << "\nEnter your choice : ";
```

```
    cin >> choice;
    switch (choice)
    {
    case 1:
        list1.create();
        break;

    case 2:
        list1.insertMember();
        break;

    case 3:
        list1.deleteMember();
        break;

    case 4:
        list1.print();
        break;

    case 5:
        list1.mergeList();
        break;

    case 6:
        cout << "\n\nNAME\tDIV\tPRN\n";
        list1.recursionPrint(list1.head);
        break;

    case 7:
        list1.count();
        break;

    case 8:
        return 0;

    default:
        cout << "Invalid choice";
        break;
    }
} while (choice != 0);
```

```
    return 0;  
}
```

Program Output:

*****MENU*****

1. Create List.
2. Insert Member
3. Delete Member
4. Print List
5. Merge List
6. Reverse List
7. Count
8. Exit

Enter your choice : 1

Enter the number of students: 3

Enter the name, division and PRN of President:

Jay 1 101

Enter the name, division and PRN of Member :

Sahil 2 102

Enter the name, division and PRN of Secretary :

Ram 1 103

*****MENU*****

1. Create List.
2. Insert Member
3. Delete Member
4. Print List
5. Merge List
6. Reverse List
7. Count
8. Exit

Enter your choice : 4

| NAMEDIV | PRN |
|---------|-----|
|---------|-----|

| | | |
|-----|---|-----|
| Jay | 1 | 101 |
|-----|---|-----|

| | | |
|-------|---|-----|
| Sahil | 2 | 102 |
|-------|---|-----|

| | | |
|-----|---|-----|
| Ram | 1 | 103 |
|-----|---|-----|

President is: Jay

Secretary is: Ram

*****MENU*****

1. Create List.
2. Insert Member
3. Delete Member
4. Print List
5. Merge List
6. Reverse List
7. Count
8. Exit

Enter your choice : 2

Enter the name, division and PRN of member to insert:

Nayan 2 104

*****MENU*****

1. Create List.
2. Insert Member
3. Delete Member
4. Print List
5. Merge List
6. Reverse List
7. Count
8. Exit

Enter your choice : 4

| NAME | DIV | PRN |
|------|-----|-----|
|------|-----|-----|

| | | |
|-----|---|-----|
| Jay | 1 | 101 |
|-----|---|-----|

| | | |
|-------|---|-----|
| Sahil | 2 | 102 |
|-------|---|-----|

| | | |
|-------|---|-----|
| Nayan | 2 | 104 |
|-------|---|-----|

| | | |
|-----|---|-----|
| Ram | 1 | 103 |
|-----|---|-----|

President is: Jay

Secretary is: Ram

*****MENU*****

1. Create List.
2. Insert Member
3. Delete Member
4. Print List
5. Merge List
6. Reverse List

7. Count

8. Exit

Enter your choice : 3

Enter the PRN of member to delete: 102

*****MENU*****

1. Create List.

2. Insert Member

3. Delete Member

4. Print List

5. Merge List

6. Reverse List

7. Count

8. Exit

Enter your choice : 4

| | NAMEDIV | PRN |
|-------|---------|-----|
| Jay | 1 | 101 |
| Nayan | 2 | 104 |
| Ram | 1 | 103 |

President is: Jay

Secretary is: Ram

*****MENU*****

1. Create List.

2. Insert Member

3. Delete Member

4. Print List

5. Merge List

6. Reverse List

7. Count

8. Exit

Enter your choice : 5

Enter the contents for second list:

Enter the number of students: 2

Enter the name, division and PRN of President:

Mohit 1 105

Enter the name, division and PRN of Secretary :

Karan 2 106

*****MENU*****

1. Create List.

2. Insert Member
3. Delete Member
4. Print List
5. Merge List
6. Reverse List
7. Count
8. Exit

Enter your choice : 4

| NAMEDIV | PRN |
|---------|-----|
|---------|-----|

| | | |
|-------|---|-----|
| Jay | 1 | 101 |
| Nayan | 2 | 104 |
| Ram | 1 | 103 |
| Mohit | 1 | 105 |
| Karan | 2 | 106 |

President is: Jay

Secretary is: Karan

*****MENU*****

1. Create List.
2. Insert Member
3. Delete Member
4. Print List
5. Merge List
6. Reverse List
7. Count
8. Exit

Enter your choice : 6

| NAMEDIV | PRN |
|---------|-----|
|---------|-----|

| | | |
|-------|---|-----|
| Karan | 2 | 106 |
| Mohit | 1 | 105 |
| Ram | 1 | 103 |
| Nayan | 2 | 104 |
| Jay | 1 | 101 |

*****MENU*****

1. Create List.
2. Insert Member
3. Delete Member
4. Print List

5. Merge List

6. Reverse List

7. Count

8. Exit

Enter your choice : 7

Total no. of students : 5

| | |
|--------------------------|---|
| Assignment Title: | Write C++ program for storing binary number using doubly linked lists. Write functions a) To compute 1's and 2's complement b) Add two binary numbers |
| Assignment No.: | A-08 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
#include <iostream>
using namespace std;
class node
{
    bool n;
    node *next, *prev;
public:
    node()
    {
        prev = NULL;
        next = NULL;
    }
    node(bool b)
    {
        n = b;
        prev = NULL;
        next = NULL;
    }
    friend class binary;
};

class binary
{
```

```
    node *start, *end;
    void prep(bool);
public:
    binary()
    {
        start = NULL;
        end = NULL;
    }
    void ones_comp();
    void twos_comp();
    void disp();
    void DtoB(int);
    int BtoD();
    binary operator+(binary);
};
```

```
void binary::DtoB(int n)
{
    int rem;
    rem = n % 2;
    node *p;
    start = new node(rem);
    end = start;
    n = n / 2;
    while (n != 0)
    {
        rem = n % 2;
        n = n / 2;
        p = new node(rem);
        start->prev = p;
        p->next = start;
        start = p;
    }
}
```

```
int binary::BtoD()
{
    node *temp = end;
    int i = 0, p = 1, s = 0;
    while (temp != NULL)
```

```
{
    if (temp->n == 1)
    {
        p = 1;
        for (int j = 1; j <= i; j++)
        {
            p *= 2;
            s += p;
        }
    }
    i += 1;
    temp = temp->prev;
}
return s;
}
```

```
void binary::disp()
{
    node *temp = start;
    while (temp != NULL)
    {
        cout << temp->n;
        temp = temp->next;
    }
}
```

```
void binary::ones_comp()
{
    node *temp = start;
    while (temp != NULL)
    {
        if (temp->n == 1)
            temp->n = 0;
        else
            temp->n = 1;
        temp = temp->next;
    }
}

void binary::twos_comp()
{
}
```

```
node *temp = end;
ones_comp();
int c = 1;
while (temp != NULL)
{
    if (temp->n == 1 && c == 1)
    {
        temp->n = 0;
        c = 1;
    }
    else if (temp->n == 1 && c == 0)
        temp->n = 1;
    else if (temp->n == 0 && c == 1)
    {
        temp->n = 1;
        c = 0;
    }
    else
        temp->n = 0;
    temp = temp->prev;
}
if (c == 1)
    prep(1);
}
```

```
void binary::prep(bool b)
{
    if (start == NULL)
    {
        start = new node(b);
        end = start;
    }
    else
    {
        node *p = new node(b);
        start->prev = p;
        p->next = start;
        start = p;
    }
}
```

```
binary binary::operator+(binary b)
{
    binary c;
    bool cry = 0;
    node *temp1 = end, *temp2 = b.end;
    while (temp1 != NULL && temp2 != NULL)
    {
        if (temp1->n == 1)
        {
            if (temp2->n == 1)
            {
                if (cry == 1)
                    c.prep(1);
                else
                {
                    cry = 1;
                    c.prep(0);
                }
            }
            else
            {
                if (cry == 1)
                    c.prep(0);
                else
                    c.prep(1);
            }
        }
        else
        {
            if (temp2->n == 1)
            {
                if (cry == 1)
                    c.prep(0);
                else
                    c.prep(1);
            }
            else
            {
                if (cry == 1)
```



```
        {
            c.prep(1);
            cry = 0;
        }
        else
            c.prep(0);
    }
}
temp1 = temp1->prev;
temp2 = temp2->prev;
}
if (temp1 == NULL)
    temp1 = temp2;
while (temp1 != NULL)
{
    if (cry == 1)
    {
        if (temp1->n == 1)
            c.prep(0);
        else
        {
            c.prep(0);
            cry = 0;
        }
    }
    else
        c.prep(temp1->n);
    temp1 = temp1->prev;
}
if (cry == 1)
    c.prep(1);
return c;
}
```

```
int main()
{
    bool f = true;
    while (f)
    {
        int ch;
```

```
int n;
binary b, d;
cout << "\n*****MENU*****";
cout << "\n1) Generate binary from decimal\n2) 1's complement\n3) 2's complement\n4) Add
2 binary numbers\n5) Exit" << endl;
cout << "\nEnter your choice: ";
cin >> ch;
cout << endl;
switch (ch)
{
case 1:
    cout << "Enter the decimal number to generate binary: ";
    cin >> n;
    b.DtoB(n);
    cout << "The binary equivalent of no is : ";
    b.disp();
    break;

case 2:
    cout << "Enter the decimal number to generate binary and display 1's complement : ";
    cin >> n;
    b.DtoB(n);
    cout << "Equivalent binary is : ";
    b.disp();
    b.ones_comp();
    cout << "\n The 1's complement of given binary number is : ";
    b.disp();
    break;

case 3:
    cout << "Enter the decimal number to generate binary and display 2's complement : ";
    cin >> n;
    b.DtoB(n);
    cout << "The binary equivalent is : ";
    b.disp();
    b.twos_comp();
    cout << "\n The 2's complement of given binary number is : ";
    b.disp();
    break;
```

```
case 4:
    cout << "Enter first decimal number : ";
    cin >> n;
    b.DtoB(n);
    cout << "The binary equivalent is : ";
    b.disp();
    cout << "\nEnter second decimal number : ";
    cin >> n;
    b.DtoB(n);
    cout << "The binary equivalent is : ";
    b.disp();
    b = b + d;
    cout << "\nThe addition of numbers is : ";
    b.disp();
    cout << " (" << b.BtoD() << ")";
    break;

case 5:
    f = false;
    break;

default:
    cout << "\n Incorrect choice";
    break;
}
}
return 0;
}
```

Program Output:

*****MENU*****

- 1) Generate binary from decimal
- 2) 1's complement
- 3) 2's complement
- 4) Add 2 binary numbers
- 5) Exit

Enter your choice: 1

Enter the decimal number to generate binary: 5

The binary equivalent of no is : 101

*****MENU*****

- 1) Generate binary from decimal
- 2) 1's complement
- 3) 2's complement
- 4) Add 2 binary numbers
- 5) Exit

Enter your choice: 2

Enter the decimal number to generate binary and diaplay 1's complement : 5

Equivalent binary is : 101

The 1's complement of given binary number is : 010

*****MENU*****

- 1) Generate binary from decimal
- 2) 1's complement
- 3) 2's complement
- 4) Add 2 binary numbers
- 5) Exit

Enter your choice: 3

Enter the decimal number to generate binary and diaplay 2's complement : 5

The binary equivalent is : 101

The 2's complement of given binary number is : 011

*****MENU*****

- 1) Generate binary from decimal
- 2) 1's complement
- 3) 2's complement
- 4) Add 2 binary numbers
- 5) Exit

Enter your choice: 4

Enter first decimal number : 5

The binary equivalent is : 101

Enter second decimal number : 5

The binary equivalent is : 101

The addition of numbers is : 1010 (10)

| | |
|--------------------------|---|
| Assignment Title: | A palindrome is a string of character that's the same forward and backward. Typically punctuation, capitalization, and spaces are ignored. For example, "Poor Dan is in a droop" is a palindrome, as can be seen by examining the characters "poor danisina droop" and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functions- a) To print original string followed by reversed string using stack b) To check whether given string is palindrome or not |
| Assignment No.: | A-09 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
#include<iostream>
#include<string.h>
#define max 50
using namespace std;
class STACK
{
    private:
        char a[max];
        int top;
    public:
        STACK()
        {
            top=-1;
        }
        void push(char);
        void reverse();
        void convert(char[]);
        void palindrome();
}
```

```
};

void STACK::push(char c)
{
    top++;
    a[top] = c;
    a[top+1]='\0';
}

void STACK::reverse()
{
    char str[max];
    cout<<"\n\nReverse string is : ";
    for(int i=top,j=0; i>=0; i--,j++)
    {
        cout<<a[i];
        str[j]=a[i];
    }
    cout<<endl;
}

void STACK::convert(char str[])
{
    int j,k,len = strlen(str);
    for(j=0, k=0; j<len; j++)
    {
        if( ( (int)str[j] >= 97 && (int)str[j] <=122 ) || ( (int)str[j] >= 65 && (int)str[j] <=90
    ))
        {
            if( (int)str[j] <=90 )
            {
                str[k] = (char)( (int)str[j] + 32 );
            }else
            {
                str[k] = str[j];
            }
            k++;
        }
    }
}
```

```
        str[k]='\0';
        cout<<endl<<"Converted String : "<<str<<"\n";
    }

void STACK::palindrome()
{
    char str[max];
    int i,j;
    for(i=top,j=0; i>=0; i--,j++)
    {
        str[j]=a[i];
    }
    str[j]='\0';
    if(strcmp(str,a) == 0)
        cout<<"\n\nString is palindrome";
    else
        cout<<"\n\nString is not palindrome";
}

int main()
{
    STACK stack;
    char str[max];
    int i=0;
    cout<<"\nEnter string to be reversed and check is it palindrome or not : \n\n";
    cin.getline(str , 50);
    stack.convert(str);
    while(str[i] != '\0')
    {
        stack.push(str[i]);
        i++;
    }
    stack.palindrome();
    stack.reverse();
}
```

| |
|------------------------|
| Program Output: |
|------------------------|

Enter string to be reversed and check is it palindrome or not :

Poor Dan is in a droop

Converted String : poordanisinadroop

String is palindrome

Reverse string is : poordanisinadroop

| | |
|--------------------------|--|
| Assignment Title: | In any language program mostly syntax error occurs due to unbalancing delimiter such as (, {, }, [,]. Write C++ program using stack to check whether given expression is well parenthesized or not. |
| Assignment No.: | A-10 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
#include<iostream>
using namespace std;
class stack
{
    char s[25];
    int top;
public:
    void push(char val);
    char pop();
    bool isempty();
    bool isfull();
    void display();
    stack()
    {
        top=-1;
    }
};

bool stack ::isempty()
{
    if(top==-1)
    {
        return -1;
    }
}
```

```
    }
    else
        return 0;
}

void stack::push(char val)
{
    if(top<=24)
    {
        top++;
        s[top]=val;
    }
    else
        cout<<"\n stack if full";
}

char stack::pop()
{
    char val;
    if(!isempty())
    {
        val=s[top];
        top--;
        return val;
    }
    else
    {
        cout<<"\n enter stack";
        return '*';
    }
}

class paranthesis
{
    char expn[25];
    stack obj;
public:
    void read();
    void checkexpn();
};
```

```
void paranthesis::read()
{
    cout<<"\n Enter the expression : ";
    cin>>expn;
    cout<<"\n Entered expression is : "<<expn;
}

void paranthesis::checkexpn()
{
    int i,flag;
    char ch;

    flag=0;
    for(i=0;expn[i]!='\0';i++)
    {
        if(expn[i]=='{' || expn[i]=='[' || expn[i]=='(')
        {
            obj.push(expn[i]);
        }
        if(expn[i]=='}' || expn[i]==']' || expn[i]==')')
        {
            if(!obj.isEmpty())
            {
                ch=obj.pop();
                if(expn[i]=='}' && ch!='{')
                {
                    flag=1;
                    break;
                }
                if(expn[i]==']' && ch!='[')
                {
                    flag=1;
                    break;
                }
                if(expn[i]==')' && ch!='(')
                {
                    flag=1;
                    break;
                }
            }
        }
    }
}
```

```
    }  
    }  
    if(flag==0 && obj.isempty())  
        cout<<"\n Expression is in well paranthesis";  
    else  
        cout<<"\n Expression is not well paranthesis";  
}  
  
int main()  
{  
    paranthesis obj1;  
    obj1.read();  
    obj1.checkexpn();  
    return 0;  
}
```

Program Output:

Enter the expression : (a+b)*c
Entered expression is : (a+b)*c
Expression is in well paranthesis

| | |
|--------------------------|---|
| Assignment Title: | Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write C++ program for simulating job queue. Write functions to add job and delete job from queue. |
| Assignment No.: | A-11 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

| |
|----------------------|
| Program Code: |
|----------------------|

```
#include<iostream>
#define max 10
using namespace std;
class Queue
{
    public:
    int front,rear,job[max];
    Queue()
    {
        front=rear=-1;
    }
    void addjob(int a);
    int deljob();
    bool is_full();
    bool is_empty();
    int getfront();
    void display();
};

bool Queue::is_empty()
{
```

```
    if(front==rear)
        return 1;
    else
        return 0;
}

bool Queue::is_full()
{
    if(rear==max-1)
        return 1;
    else
        return 0;
}

void Queue::addjob(int a)
{
    if(is_full())
        cout<<"Queue is full.";
    else
        job[++rear]=a;
}

int Queue::deljob()
{
    if(is_empty())
        cout<<"Queue is empty.";
    else
        return(job[++front]);
}

int Queue::getfront()
{
    if(is_empty())
        cout<<"Queue is empty.";
    else
        cout<<(job[front+1]);
}

void Queue::display()
{
```

```
int i;
for(i=front+1;i<=rear;i++)
    cout<<job[i]<<" ";
}

int main()
{
    Queue q;
    int ch,x;
    do
    {
        cout<<"\n*****MENU*****";
        cout<<"\n 1.Insert job\n 2.Delete job\n 3.Display\n 4.Exit\n Enter your choice :";
        cin>>ch;
        switch(ch)
        {
            case 1:
                cout<<"\n Enter job :";
                cin>>x;
                q.addjob(x);
                break;
            case 2:
                int A;
                A=q.deljob();
                cout<<"\n Deleted job :"<<A;
                break;
            case 3:
                cout<<"\n Queue contains :";
                q.display();
                break;
            case 4:
                cout<<"\nExiting program.";
        }
    }while(ch!=4);
    return 0;
}
```

| |
|------------------------|
| Program Output: |
|------------------------|

*****MENU*****

1.Insert job

2.Delete job

3.Display

4.Exit

Enter your choice :1

Enter job :101

*****MENU*****

1.Insert job

2.Delete job

3.Display

4.Exit

Enter your choice :1

Enter job :102

*****MENU*****

1.Insert job

2.Delete job

3.Display

4.Exit

Enter your choice :1

Enter job :103

*****MENU*****

1.Insert job

2.Delete job

3.Display

4.Exit

Enter your choice :3

Queue contains :101 102 103

*****MENU*****

1.Insert job

2.Delete job

3.Display

4.Exit

Enter your choice :2

Deleted job :101

*****MENU*****

1.Insert job

2.Delete job

3.Display

4.Exit

Enter your choice :3

Queue contains :102 103

*****MENU*****

1.Insert job

2.Delete job

3.Display

4.Exit

Enter your choice :4

Exiting program.

| | |
|--------------------------|--|
| Assignment Title: | A double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. Obtain a data representation mapping a deque into a one-dimensional array. Write C++ program to simulate deque with functions to add and delete elements from either end of the deque. |
| Assignment No.: | A-12 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
#include<iostream>
#include<stdio.h>
#define MAX 10
using namespace std;
struct que
{
int arr[MAX];
int front,rear;
};

void init(struct que *q)
{
q->front=-1;
q->rear=-1;
}

void print(struct que q)
{
int i;
i=q.front;
while(i!=q.rear)
```

```
{
cout<<"\t"<<q.arr[i];
i=(i+1)%MAX;
}
cout<<"\t"<<q.arr[q.rear];
}

int isempty(struct que q)
{
return q.rear==-1?1:0;
}

int isfull(struct que q)
{
return (q.rear+1)%MAX==q.front?1:0;
}

void addf(struct que *q,int data)
{
if(isempty(*q))
{
q->front=q->rear=0;
q->arr[q->front]=data;
}
else
{
q->front=(q->front-1+MAX)%MAX;
q->arr[q->front]=data;
}
}

void addr(struct que *q,int data)
{
if(isempty(*q))
{
q->front=q->rear=0;
q->arr[q->rear]=data;
}
else
{

```

```
q->rear=(q->rear+1)%MAX;
q->arr[q->rear]=data;
}
}
```

```
int delf(struct que *q)
{
int data1;
data1=q->arr[q->front];
if(q->front==q->rear)
init(q);
else
q->front=(q->front+1)%MAX;
return data1;
}
```

```
int delr(struct que *q)
{
int data1;

data1=q->arr[q->rear];
if(q->front==q->rear)
init(q);
else
q->rear=(q->rear-1+MAX)%MAX;
return data1;
}
```

```
int main()
{
struct que q;
int data,ch;
init(&q);
while(ch!=6)
{
cout<<"\n*****MENU*****";
cout<<"\n1. Insert at beginning";
cout<<"\n2. Insert at end";
cout<<"\n3. Deletion from front";
cout<<"\n4. Deletion from rear";
```

```
cout<<"\n5. Display";
cout<<"\n6. Exit";
cout<<"\n\nEnter your choice : ";
cin>>ch;
switch(ch)
{
case 1:
cout<<"\nEnter data to insert front : ";
cin>>data;
addf(&q,data);
break;
case 2:
cout<<"\nEnter the data to insert rear : ";
cin>>data;
addr(&q,data);
break;
case 3:
if(isempty(q))
cout<<"\nDeque is empty!!!";
else
{
data=delf(&q);
cout<<"\nDeleted data is : "<<data;
}
break;
case 4:
if(isempty(q))
cout<<"\nDeque is empty!!!";
else
{
data=delr(&q);
cout<<"\nDeleted data is : "<<data;
}
break;
case 5:
if(isempty(q))
cout<<"\nDeque is empty!!!";
else
{
cout<<"\nDeque elements are : ";
```

```
print(q);
}
break;
}
}
return 0;
}
```

Program Output:

*****MENU*****

1. Insert at beginning
2. Insert at end
3. Deletion from front
4. Deletion from rear
5. Display
6. Exit

Enter your choice : 1

Enter data to insert front : 101

*****MENU*****

1. Insert at beginning
2. Insert at end
3. Deletion from front
4. Deletion from rear
5. Display
6. Exit

Enter your choice : 2

Enter the data to insert rear : 102

*****MENU*****

1. Insert at beginning
2. Insert at end
3. Deletion from front
4. Deletion from rear
5. Display
6. Exit

Enter your choice : 1

Enter data to insert front : 103

*****MENU*****

1. Insert at beginning
2. Insert at end
3. Deletion from front
4. Deletion from rear
5. Display
6. Exit

Enter your choice : 2

Enter the data to insert rear : 104

*****MENU*****

1. Insert at beginning
2. Insert at end
3. Deletion from front
4. Deletion from rear
5. Display
6. Exit

Enter your choice : 5

Deque elements are : 103 101 102 104

*****MENU*****

1. Insert at beginning
2. Insert at end
3. Deletion from front
4. Deletion from rear
5. Display
6. Exit

Enter your choice : 3

Deleted data is : 103

*****MENU*****

1. Insert at beginning
2. Insert at end
3. Deletion from front
4. Deletion from rear
5. Display
6. Exit

Enter your choice : 4

Deleted data is : 104

*****MENU*****

1. Insert at beginning
2. Insert at end
3. Deletion from front
4. Deletion from rear
5. Display
6. Exit

Enter your choice : 5

Deque elements are : 101 102

| | |
|--------------------------|--|
| Assignment Title: | Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using array. |
| Assignment No.: | A-13 |
| Student Name: | Shermale Vaibhav Valmik |
| Year & DIV.: | SE A |
| Batch: | D |
| Roll No: | 64 |

Program Code:

```
#include<iostream>
using namespace std;
#define max 20
class cque
{
    public:
    int front,rear,que[max];
    cque()
    {
        front=rear=-1;
    }
    void add(int a);
    void delorder();
    int is_full();
    int is_empty();
    void display();
};

int cque::is_full()
{
    if(((front==0)&&(rear==(max-1)))||(front==(rear+1)))
        return 1;
    else
```

```
        return 0;
    }

int cque::is_empty()
{
    if(front==-1)
        return 1;
    else
        return 0;
}

void cque::add(int a)
{
    if(is_full())
        cout<<"Queue is full.";
    else if(front==-1)
        front=rear=0;
    else
    {
        rear=(rear+1)%max;
        que[rear]=a;
    }
}

void cque::delorder()
{
    int i;
    i=que[front];
    if(front==rear)
        front=rear=-1;
    else
    {
        front=(front+1)%max;
    }
    cout<<"\nOrder deleted : "<<i;
}

void cque::display()
{
    int temp;
```

```
        temp=front;
        if(is_empty())
            cout<<"Queue is empty.";
        else
        {
            while(temp!=rear)
            {
                cout<<" "<<que[temp];
                temp=(temp+1)%max;
            }
            cout<<" "<<que[temp];
        }
    }
}

int main()
{
    cque c;
    int ch,x;
    do
    {
        cout<<"\n1.Insert order\n2.Delete order\n3.Display orders\n4.Exit\nEnter your
choice:";
        cin>>ch;
        switch(ch)
        {
            case 1:
                cout<<"\nEnter order no.:";
                cin>>x;
                c.add(x);
                break;
            case 2:
                c.delorder();
                break;
            case 3:
                cout<<"Queue contains :";
                c.display();
                break;
            case 4:
                cout<<"\nExiting program..";
        }
    }
```

```
    }while(ch!=4);  
    return 0;  
}
```

Program Output:

```
1. Insert order  
2..Delete order  
3..Display orders  
4..Exit  
Enter your choice:1  
Enter order no.:102  
1..Insert order  
2..Delete order  
3..Display orders  
4..Exit  
Enter your choice:1  
Enter order no.:103  
1..Insert order  
2..Delete order  
3..Display orders  
4..Exit  
Enter your choice:1  
Enter order no.:101  
1..Insert order  
2..Delete order  
3..Display orders  
4..Exit  
Enter your choice:3  
Queue contains : 102 103 101  
1..Insert order  
2..Delete order  
3..Display orders  
4..Exit  
Enter your choice:2  
Order deleted :102  
1..Insert order  
2..Delete order  
3..Display orders
```

4..Exit

Enter your choice:4

Exiting program..